



# UNIVERSIDAD DE LA RIOJA

## TRABAJO FIN DE ESTUDIOS

Título

Oferta y asignación de prácticas universitarias en empresa

Autor/es

GUILLERMO LOBERA GARCIA

Director/es

ARTURO JAIME ELIZONDO

Facultad

Facultad de Ciencia y Tecnología

Titulación

Grado en Ingeniería Informática

Departamento

MATEMÁTICAS Y COMPUTACIÓN

Curso académico

2019-20



***Oferta y asignación de prácticas universitarias en empresa***, de GUILLERMO LOBERA GARCIA

(publicada por la Universidad de La Rioja) se difunde bajo una Licencia Creative Commons Reconocimiento-NoComercial-SinObraDerivada 3.0 Unported.

Permisos que vayan más allá de lo cubierto por esta licencia pueden solicitarse a los titulares del copyright.

© El autor, 2020

© Universidad de La Rioja, 2020

[publicaciones.unirioja.es](http://publicaciones.unirioja.es)

E-mail: [publicaciones@unirioja.es](mailto:publicaciones@unirioja.es)



**UNIVERSIDAD  
DE LA RIOJA**

Facultad de Ciencia y Tecnología

# **TRABAJO FIN DE GRADO**

Grado en Ingeniería Informática

Oferta y asignación de  
prácticas universitarias en empresa

-

Offer and assignment  
of university practices in companies

Realizado por:

Guillermo Lobera García

Tutelado por:

Arturo Jaime Elizondo

**Logroño, Junio de 2020**

## Resumen

Este trabajo se encuadra en un proyecto de empresa que desarrolló una aplicación web a una universidad para gestionar la oferta y asignación de las prácticas en empresa de sus estudiantes. Entre los tipos de usuario destacan los tutores de prácticas de las empresas y los administradores de la universidad. Las tareas realizadas como trabajo de fin de grado incluyen el desarrollo de la funcionalidad asociada a varias páginas de la aplicación, la revisión y documentación de los requisitos de la aplicación completa y la verificación final de la misma. La última actividad contó con la participación activa del cliente. En la verificación se depuraron errores y se incorporaron mejoras sobre la implementación del equipo. También se participó en la implantación de varias versiones del producto. El equipo del proyecto constaba de tres personas de Logroño, dos de Mallorca y una de Zaragoza. Los clientes residían en Pamplona. Por tanto, la telecolaboración ha sido constante durante la vida del proyecto y del trabajo fin de grado.

El método de trabajo ha sido iterativo e incremental, incluyendo el despliegue de las sucesivas versiones en servidores del cliente. La interacción con el cliente ha sido constante. La arquitectura sigue el patrón Modelo-Vista-Controlador. Para el *frontend* se ha usado el framework Angular, para el *backend* .Net y una API REST para su interacción. El *backend* accede mediante servicios HTTP a los datos residentes en la ERP Navision del cliente.

## Abstract

This project is part of a company project where the goal was to develop a web application to allow an university to manage the offer and assignment of its students in practices of companies. There are many types of users in the application, emphasizing on company tutors and university administrators. The tasks encompass developing the functionality of several pages of the application, reviewing and documenting the requirements of the whole application and verifying its result, with the active participation of the client. Debugging and implementing many upgrades were the main point of the verification. The deployment of the different versions of the application was done as part of this project. The development team consisted of three members from Logroño, two from Mallorca and one from Zaragoza. This, with the client being from Pamplona, has meant a constant remote cooperation during the whole development of the project and this end-of-degree work.

The work method has been iterative and incremental, including the deployment of the consecutive versions in the servers of the client. The interaction with the client has been constant. The architecture used is Model-View-Controller. The Angular framework has been utilised to develop the *frontend* of the application, and the .NET Core framework for the *backend*. Both communicate through an API REST. The *backend* gains access to the data of the application by using the HTTP services provided by the ERO Navision of the client.

# Índice

<b>Introducción</b>	<b>4</b>
Contexto	4
Motivación	4
<b>Antecedentes</b>	<b>5</b>
<b>Alcance y objetivos</b>	<b>6</b>
<b>Gestión</b>	<b>6</b>
Alcance (requisitos)	7
Cronograma	7
Recursos (dedicaciones)	8
<b>Método de desarrollo</b>	<b>8</b>
<b>Análisis y diseño</b>	<b>9</b>
Proceso seguido	9
Diseño de la interfaz	10
Funcionalidad principal	10
Requisitos no funcionales	12
Arquitectura	12
Diseño de datos	14
<b>Implementación y verificación</b>	<b>14</b>
Tecnologías y herramientas	15
Reglas de validación de datos	15
Errores del backend	16
Multilingüismo	16
Verificación y depuración	17
Una página especial: “modo consulta”	18
<b>Implantación</b>	<b>18</b>
<b>Conclusiones</b>	<b>18</b>

# 1. Introducción

En este apartado justificamos la elección del proyecto en el que se enmarca el trabajo de fin de grado (TFG). Comenzaremos contextualizando el trabajo y explicando la situación por la que el cliente decidió poner en marcha el proyecto. Después explicaremos los motivos que nos llevaron a realizarlo. Presentaremos cómo se organiza la memoria distinguiendo qué partes del proyecto se han abordado en el TFG.

## 1.1. Contexto

El presente TFG es una parte del proyecto desarrollado por la empresa Hiberus-Osaba a un cliente concreto: la universidad pública de Navarra (UPNA). El proyecto tiene como objetivo automatizar los procesos de oferta y asignación de las prácticas que realizan los estudiantes universitarios en empresas de diferentes sectores relacionados con las titulaciones que se imparten en esta institución académica.

La gestión que se venía haciendo desde la universidad era de tipo manual. Esto incluye atender a las ofertas de prácticas de las empresas, publicitarlas entre los estudiantes de las titulaciones relacionadas y realizar las gestiones administrativas correspondientes, como firma de convenios, seguros, etc. Realizar estos procesos manualmente, considerando la cantidad de comunicaciones necesarias con los interesados (por email o teléfono), exige la inversión de mucho tiempo tanto a la universidad como a las propias empresas. Estas deben ser conscientes de la forma de contactar, el periodo de realización de las ofertas, y hacer después el seguimiento de las mismas. En este contexto, disponer de una solución que centralice tanto los procesos administrativos, como la comunicación entre interesados y la información necesaria, puede conllevar la reducción en la carga de trabajo de la administración universitaria y un punto de referencia para las empresas interesadas en contribuir con sus ofertas de prácticas.

La Universidad y empresa alcanzaron un acuerdo para la elaboración de una aplicación web que tratase de mejorar la situación anterior. La aplicación deberá permitir a las empresas ofertar prácticas y hacerles seguimiento, y la universidad incorporar información de los estudiantes interesados y controlar el proceso de asignación de prácticas, entre otras cuestiones administrativas.

## 1.2. Motivación

Después de realizar las prácticas curriculares en la mencionada empresa, me ofrecieron algunas alternativas de TFG. Esta propuesta tenía varios puntos fuertes, el primero de ellos era que ya había participado en algunas actividades del proyecto en el periodo de prácticas. Además, me había estado formando en las principales tecnologías que se utilizaban, con lo que era una oportunidad para asentar y extender lo aprendido. Se utilizaban tecnologías de desarrollo de aplicaciones web modernas y muy utilizadas (.Net, Angular, etc.). Se trataba además de un proyecto en el que iba a colaborar con un conjunto de profesionales con experiencia, desarrollando un producto para un cliente externo que lo necesitaba tener

disponible para una fecha cercana a la finalización del TFG. Por tanto iba a participar en la fase final del proyecto y en su implantación final.

En conclusión, se trataba de un proyecto asumible, interesante, integrado en un equipo *full stack* (programadores senior con experiencia), integrado por subequipos virtuales o remotos, bien dimensionado respecto al alcance de un TFG, y con cierto contacto con el cliente. Una experiencia con un proyecto real con toda la problemática a superar incluyendo la comunicación con el cliente.

Esta memoria la organizamos de la siguiente manera. Comenzamos explicando los antecedentes del proyecto en el que se enmarca el TFG. A continuación se describen los objetivos del TFG y un resumen del alcance. Le sigue el capítulo de gestión. Se trata de una gestión post-mortem a excepción de los requisitos que ya estaban recogidos antes del inicio del TFG (aunque fue necesario, e importante, documentarlos, revisarlos y completarlos). A esto le sigue una sección breve sobre la metodología de trabajo, donde la comunicación con los miembros virtuales ha forzado a utilizar comunicaciones telemáticas desde el principio. Tras describir brevemente la arquitectura se presentan los requisitos funcionales abordados en el TFG. También se revisa brevemente la estructura de los datos utilizados, cuyo diseño ya existía antes de comenzar el TFG.

En este TFG distinguimos dos fases. La primera es el diseño e implementación de algunas de las páginas web que recogen parte de la funcionalidad de la aplicación. Dentro de esta fase se incluye el despliegue de sucesivas versiones. La segunda es la revisión total de la aplicación como consecuencia del proceso de verificación realizado con la participación del cliente y la implantación final. El apartado de implementación y verificación destaca los problemas más interesantes encontrados durante estas dos fases y cómo se resolvieron.

La memoria finaliza con una sección de conclusiones donde se destacan los puntos relevantes de la experiencia y se trata de extraer alguna lección aprendida.

## 2. Antecedentes

La empresa donde se ha desarrollado el proyecto está especializada en servicios de consultoría de negocio, desarrollo tecnológico y outsourcing. Las prácticas curriculares se realizaron en la misma empresa. Ese periodo se dedicó a la formación e intervención directa en el desarrollo de aplicaciones web. Se profundizó en algunas tecnologías y arquitecturas que han acabado utilizándose en el TFG. Destaco las siguientes:

- Framework Angular para las interfaces (*frontend*) de aplicaciones web.
- Framework .NET Core para la lógica (*backend*) de aplicaciones web.
- Uso de lenguajes de programación TypeScript, HTML y C#.
- Arquitectura REST para la creación de servicios web.
- Estructura de software Modelo-Vista-Controlador.

Durante el periodo de prácticas, la empresa ya estaba desarrollando un proyecto web para una universidad, cuyo objetivo era el desarrollo de una herramienta web para la organización de las prácticas de empresa de los estudiantes. En ese momento participé en la elaboración de un prototipo inicial de la herramienta, que se presentó al cliente. El prototipo se puede consultar en el Anexo A de esta memoria. El TFG se ha enmarcado en la etapa final del mencionado proyecto.

La herramienta web necesitaba acceder a datos que la universidad cliente almacena en bases de datos de su ERP Navision. La solución web debe ofrecer funcionalidades dirigidas a las empresas, en las que los estudiantes realizan sus prácticas, para incorporar y gestionar su oferta formativa. Esta actividad de las empresas debería reflejarse en la base de datos de la ERP Navision de la universidad. La universidad, por su parte, contaría con la funcionalidad asociada a las prácticas desde el punto de vista de la administración universitaria.

En el desarrollo del proyecto de gestión de prácticas de empresa han participado varios interesados. Del cliente-universidad, intervino el personal de su servicio informático, quienes debían proporcionar los servicios apropiados para el acceso a su ERP desde la herramienta a desarrollar. Además han colaborado dos equipos de la empresa sitos en dos lugares geográficos distantes como son La Rioja y Mallorca. Además, durante la ejecución del proyecto también se incorporó al proyecto otra persona desde Zaragoza. El coordinador del proyecto ha sido mi tutor de empresa de prácticas curriculares y del TFG, Íñigo León.

Antes de comenzar el TFG, el cliente aceptó la propuesta recogida en el prototipo mencionado anteriormente. El equipo mallorquín fue el que comenzó el desarrollo del proyecto avanzando en su programación hasta completar aproximadamente el 60% del alcance previsto. Al inicio de este TFG quedaban por implementar varias páginas web, y la corrección de errores detectados en las páginas desarrolladas en Mallorca. También faltaba realizar la verificación general del producto en estrecha colaboración con el cliente y atender a los cambios y correcciones que éste pudiera solicitar.

### 3. Alcance y objetivos

El objetivo general del proyecto en el que se enmarca el TFG es el desarrollo de una herramienta online para la gestión administrativa de las prácticas en empresa de una universidad. La solución cuenta con tres tipos de usuario. Los primeros son tutores de prácticas de empresa, que controlan su oferta de prácticas y empleo. Los segundos son administrativos de las empresas con permiso para agregar contactos de la empresa. Por último, administrativos de la universidad que pueden consultar a toda la información de la aplicación, pero no pueden modificarla. El producto resultante del proyecto se debe desplegar en un servidor Microsoft IIS del cliente.

En el TFG se ha realizado una parte de este proyecto. Podemos distinguir dos fases, una de desarrollo y otra de verificación con los clientes universitarios y la posterior puesta en marcha de la solución. La fase de desarrollo se centra en las páginas desde las que se realiza el cambio de contraseña de cualquier usuario de la aplicación, la que permite a las empresas crear sus ofertas de prácticas y por último una pantalla que permite a los administradores universitarios la consulta de los datos, personas de contacto y ofertas de las empresas. La verificación, sin embargo, se extiende a toda la aplicación.

### 4. Gestión

La siguiente información que se muestra acerca de la gestión del proyecto ha sido plasmada al final del proyecto como un análisis post-mortem. La razón principal de hacerlo así es que



no se ha realizado nunca un proyecto similar que nos permita planificar las tareas con criterio y que la dirección del proyecto en el que se enmarca el TFG está en manos de otra persona.

## 4.1. Alcance (requisitos)

En la figura 3.1 se muestra la descomposición en tareas del TFG. El paquete de trabajo (nodo hoja) etiquetado con “TFG.30 Otros” representa el despliegue de la aplicación en los servidores del cliente. El resto de etiquetas de paquetes de trabajo identifica los entregables a obtener.

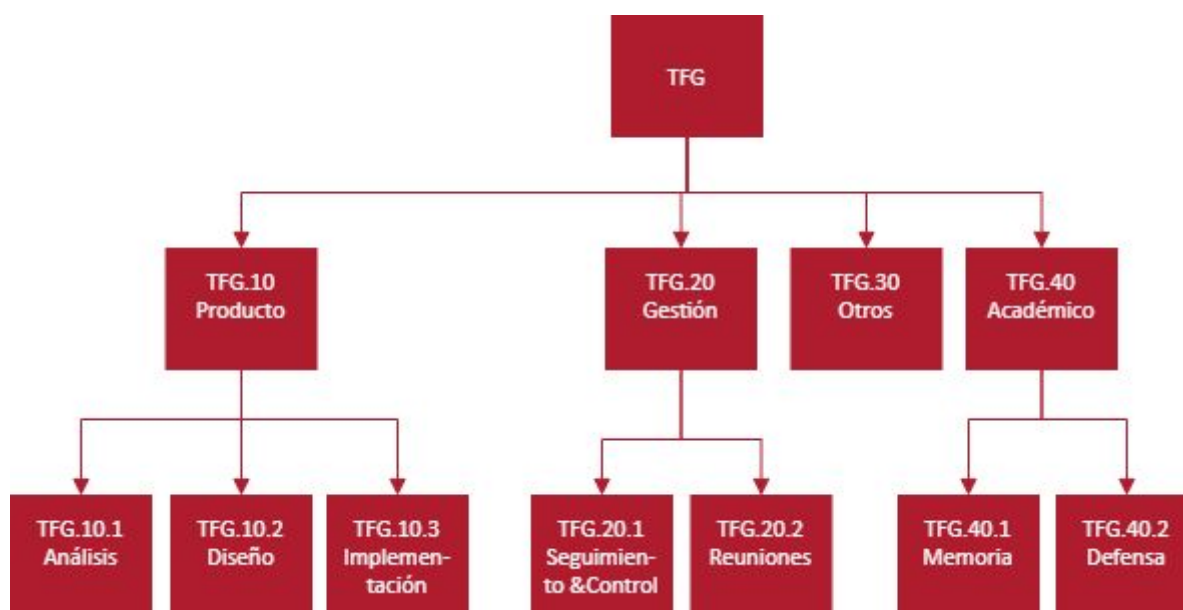


Figura 3.1. EDT del proyecto.

## 4.2. Cronograma

La figura 4.1 contiene los hitos del TFG. La figura 4.2 es el cronograma asociado a la EDT mostrada anteriormente, con los periodos de tiempo en el que se realizaron las tareas.

Hito	Febrero				Marzo				Abril				Mayo				Junio				Julio		
	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23
Inicio TFG		◆																					
Finalizar Fase 1								◆															
Desplegar Fase 1								◆															
Finalizar Fase 2									◆														
Desplegar Fase 2									◆														
Finalizar Fase 3										◆													
Desplegar Fase 3										◆													
Finalizar Fase 4											◆												
Desplegar Fase 4												◆											
Memoria revisar																			◆				
Depósito																					◆		
Defensa TFG																							◆

Figura 4.1. Diagrama de hitos principales del proyecto.

Periodo ejecución paquete TFG	Febrero				Marzo					Abril				Mayo				Junio				Julio		
	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21	S22	S23	S24
TFG.10.1. Análisis																								
TFG.10.2. Diseño																								
TFG.10.3. Implementac.																								
TFG.20.1. Segu&control																								
TFG.20.2. Reuniones																								
TFG.40.1. Memoria																								
TFG.40.2. Defensa																								

Figura 4.2. Cronograma del proyecto.

### 4.3. Recursos (dedicaciones)

En la figura 4.3 se muestran las horas semanales dedicadas a cada uno de los paquetes de trabajo de la EDT del proyecto (en cada paquete de trabajo se obtiene un entregable)..

Dedicación en horas paquete TFG	Febrero				Marzo					Abril				Mayo				Junio				TOTAL		
	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	S13	S14	S15	S16	S17	S18	S19	S20	S21		S22	
10.1. Análisis			1	1	1	1	1	1	1	1	1	1	1	1	1	1							14	
10.2. Diseño														2	2	2							6	
10.3. Implementación			21	20	21	20	21	20	21	20	12	15	20	18	19	18							266	
20.1. Segu&control				1		1		1		1		1		1		1							7	
20.2. Reuniones			1	1	1	1	1	1	1	1	1	1	1	1	1	1							14	
30.1. Memoria											5	5	5	5	5	5	5	5	2	2	1		50	
30.2. Defensa																					4	4	8	
Total.			23	23	23	23	23	23	23	23	19	23	28	28	28	28	5	5	2	2	5	4		365

Figura 4.3. Horas de dedicación a cada paquete por semana redondeadas a horas enteras.

## 5. Método de desarrollo

Durante la realización del proyecto, el equipo de desarrollo siguió un proceso iterativo e incremental. Las diferentes versiones permitían navegar por toda la aplicación permitiendo utilizar únicamente la funcionalidad ya desarrollada. Estas versiones se fueron implantando en los servidores del cliente para que pudiera experimentar. En cada iteración se fue añadiendo la funcionalidad programada y su utilización permitió detectar y depurar errores de implementación y se iban aprobando y añadiendo los cambios solicitados por el cliente.

En el proyecto colaboraron equipos de desarrollo situados en tres lugares geográficos distantes. Además, el propio cliente estaba en otra ciudad, con lo que la comunicación telemática ha sido imprescindible.

Se utilizó Google Drive para compartir y comentar la documentación del proyecto, Skype y el email corporativo como medios de mensajería asíncrona, y Whereby como plataforma de videoconferencia. En estas reuniones participamos los 3 integrantes del equipo de desarrollo de Logroño y los 2 integrantes de Mallorca, de los cuales permaneció solamente uno para continuar el proyecto.

A partir de marzo, se adquirieron licencias de Microsoft Teams, y la comunicación se centralizó allí. Se usó el chat, el sistema de videoconferencias y el sharepoint-onedrive, integrado en el propio Teams, como nuevo sistema de información en sustitución de Drive. Allí se reubicó toda la documentación compartida por el equipo. Además, se continuó usando el email corporativo.

El cambio a Teams no fue consecuencia de la ineficacia de los medios que se venían utilizando hasta entonces. El objetivo de adquirir las licencias era centralizar las comunicaciones, incluyendo el sistema de información, de este y otros proyectos remotos de la empresa. Esta centralización simplificó la comunicación, haciéndola más cómoda. Se creó un “equipo” en la herramienta que integraba a todos los desarrolladores. La documentación se asocia y comparte automáticamente para el equipo y se accede a través de la pestaña (archivos).

Se organizaron videoconferencias diarias con Teams a la misma hora de la mañana. Aunque se hacían sin webcam, se mostraban documentos y también se utilizaba la pizarra compartida. Las reuniones duraban un máximo de 20 minutos y no tenían orden del día previo. En ellas se compartían los avances logrados y las dudas que habían surgido. Se estableció un recordatorio con el planificador de Teams que avisaba diariamente del comienzo de la reunión. En estas reuniones participamos los tres desarrolladores de Logroño, el de Zaragoza y uno de Mallorca. También se unía una persona del servicio informático de la universidad y en algunas ocasiones también el representante del cliente, para asistir a demostraciones de las funcionalidades que se iban añadiendo a la aplicación. Parte de esta comunicación con los clientes formó parte del trabajo realizado en el TFG.

La comunicación por chat, entre los mismos interesados, fue constante y su uso principal fue anticipar la resolución de dudas que surgían durante la codificación (evitando esperar a la siguiente reunión). El correo electrónico fue el medio de comunicación habitual con el representante del cliente y se utilizó para que nos aclarara dudas. También se utilizó para compartir por escrito los acuerdos adoptados e informar de los avances tratados en las reuniones diarias. Estos fueron especialmente útiles para quienes se ausentaron de alguna reunión.

## 6. Análisis y diseño

### 6.1. Proceso seguido

Cuando comenzó el TFG, los desarrolladores del proyecto manejaban un documento de análisis incompleto. Así que una de la primeras tareas del trabajo fue completarlo a partir de las informaciones transmitidas previamente por el cliente. Los requisitos funcionales de dicho documento se recogen en el anexo B. También se había elaborado el prototipo incluido en el anexo A, que contaba con el visto bueno del cliente. Una parte de ese prototipo formó parte de la tarea realizada durante las prácticas curriculares, concretamente las figuras 1.1 a 1.12 y desde la 1.18 en adelante.

Además, al inicio del TFG ya estaba decidida la arquitectura, el diseño de interfaces, servicios y datos, así como las tecnologías a utilizar. En las siguientes subsecciones se presenta el resultado obtenido del proceso de análisis y diseño, diferenciando la funcionalidad desarrollada en el ámbito del TFG.

## 6.2. Diseño de la interfaz

La aplicación web que se estaba desarrollando consta de diferentes páginas cuyas imágenes se pueden consultar en el prototipo, aprobado por el cliente, del anexo A.

Las siguientes páginas son de acceso público, es decir, no precisan un inicio de sesión:

- **Inicio de sesión:** pide usuario y contraseña.
- **Dar de alta empresa:** pide el NIF de la empresa y la da de alta. Primero verifica que el NIF no existe y después permite aportar los datos de la empresa.
- **Dar de alta un contacto:** se crea un contacto que hará de administrador de la empresa, que deberá haber sido creada previamente.
- **Resetear contraseña:** se solicita una nueva contraseña para un usuario y correo electrónico concretos.

Las siguientes páginas exigen el inicio previo de sesión:

- **Pantalla maestra:** es el punto de inicio desde donde se accede al resto de páginas.
- **Ver detalles contacto:** muestra los datos de un contacto.
- **Cambio de contraseña.**
- **Empresa:** muestra los datos de la empresa a la que pertenece el usuario. Desde aquí se puede navegar a *modificar datos de empresa*.
- **Modificar datos de empresa:** se editan los datos de la empresa del usuario.
- **Modificar datos de usuario:** se editan los datos del usuario.
- **Gestionar contactos:** muestra el listado de usuarios de la empresa a la que pertenece el usuario de la sesión. El listado contiene información descriptiva de los usuarios. Desde aquí se puede navegar a *dar de alta un contacto* y a *modificar datos de usuario*. También permite eliminar un contacto de la lista.
- **Mis ofertas:** muestra el listado de las prácticas ofrecidas por la empresa del usuario de la sesión. Se pueden filtrar por estado y fecha de caducidad. Desde aquí se puede navegar a *crear una nueva oferta*.
- **Crear una nueva oferta:** hay cuatro tipos de práctica, las curriculares, las extracurriculares, las de talento y las de empleo. Hay un formulario para cada tipo.
- **Mis ofertas:** se editan los datos de una oferta de prácticas. También se puede duplicar una oferta para construir otra similar. Desde aquí se puede navegar a *gestionar candidatos* para una oferta.
- **Gestionar candidatos:** muestra los candidatos asociados a una oferta de prácticas, y permite modificar su estado respecto a la oferta y descargar su currículum. Si la oferta es extracurricular o talento permite navegar a *editar convenio*.
- **Editar convenio:** permite incluir un convenio específico para prácticas de tipo extracurricular o talento.
- **Modo consulta:** permite acceder a los datos de otra empresa, pero sin poder modificarlos. Únicamente accesible a los usuarios con un rol muy específico.

## 6.3. Funcionalidad principal

El anexo B recopila los requisitos funcionales de la aplicación completa. En esta sección se van a presentar de manera resumida las funcionalidades en las que se ha intervenido de una forma más extensa. La funcionalidad restante se ha tenido que revisar en la etapa de

depuración de errores detectados en el producto en desarrollo (principalmente por parte de los miembros mallorquines del equipo) y en la verificación final de la aplicación.

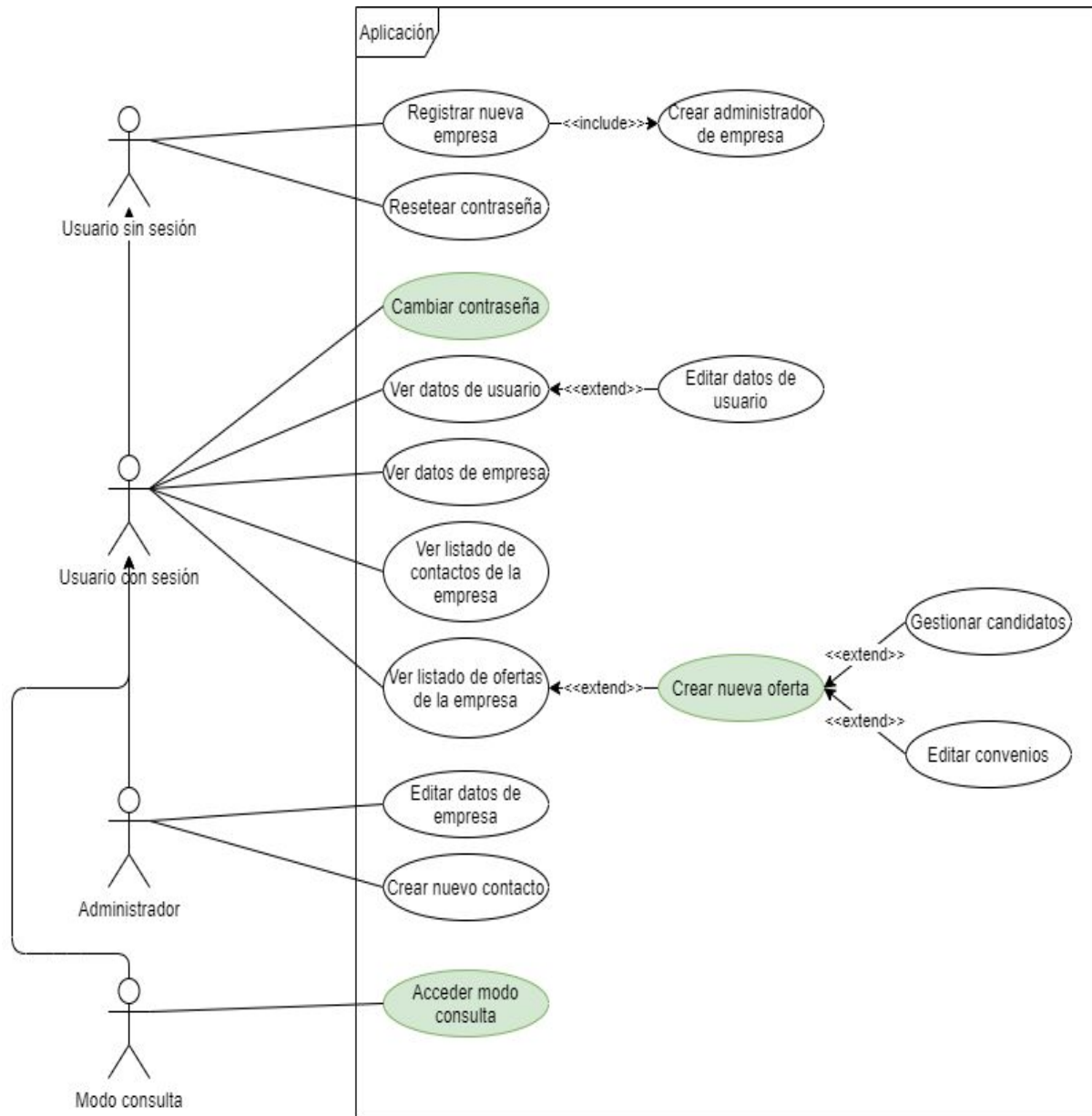


Figura 6.1: Diagrama de casos de uso. Se destacan en verde los desarrollados en el TFG.

Una de las páginas en las que se ha intervenido completamente en el TFG es la de **cambiar contraseña**. Tiene la funcionalidad habitual de este tipo de página. Sólo es accesible cuando se ha iniciado sesión, requiere que se introduzca previamente la contraseña que se pretende cambiar, y pide la verificación de la nueva contraseña propuesta.

También se ha desarrollado en el TFG desde cero la página **modo consulta**. Aunque al inicio del proyecto se planteó su necesidad, más adelante se descartó por la complejidad que entrañaba su desarrollo respecto a su utilidad práctica. Sin embargo, al final del proyecto se retomó la cuestión y se diseñó una solución menos ambiciosa pero satisfactoria para el cliente. A esta página pueden acceder únicamente usuarios con todos los permisos de la aplicación. Permite elegir una de las empresas registradas y consultar toda su información sobre contactos y ofertas. Está pensada para verificar errores detectados por usuarios o resolver dudas, pero es una página de consulta, desde la que no se pueden modificar datos

de las empresas ni de las ofertas. Por eso, aunque estos usuarios puedan, por ejemplo, acceder a las ofertas de una empresa, no tienen posibilidad de crear nuevas ofertas.

Por último, dentro del TFG se ha desarrollado la mayor parte de la funcionalidad asociada a **crear una nueva oferta**. Su objetivo es permitir crear las ofertas de prácticas en empresa, que pueden ser de cuatro tipos (curriculares, extracurriculares, de talento y de empleo). Hay bastantes datos comunes a todos los tipos de oferta y algunos especializados en uno de los tipos.

En la Figura 6.1 se muestra el Diagrama de Casos de Uso de la aplicación y se han marcado aquellos que han sido desarrollados totalmente en el TFG o con una participación muy alta. Se distinguen varios roles con permisos diferentes. Hay usuarios de acceso público y con sesión y entre estos últimos están los usuarios pertenecientes a una empresa (“Usuario con sesión”), los administradores de las empresas (“Administrador”), y los administradores de la universidad con el rol especial “Modo consulta” con permiso de consulta de todos los datos de las empresas.

## 6.4. Requisitos no funcionales

Distinguimos tres tipos de requisito no funcional:

1. *Multiplataforma*. La aplicación estará disponible online y deberá ser accesible desde cualquier dispositivo con cualquier sistema operativo, que tenga instalado uno de los navegadores más habituales (Chrome, Firefox, Edge o Safari).
2. *Escalabilidad*. La elaboración de futuros módulos con las mismas tecnologías que las usadas en el desarrollo principal deben resultar factibles y sencillas de incorporar.
3. *Accesibilidad*. La aplicación debe poder ser utilizable por usuarios con dificultades visuales sin tener que realizar un esfuerzo extra para ello.

## 6.5. Arquitectura

Se utiliza la arquitectura habitual de tres capas siguiendo el patrón modelo-vista-controlador (MVC). Como es bien conocido, el *modelo* contiene los datos a mostrar, la *vista* es la información enviada al cliente y los mecanismos de interacción con dicha información. Por último el *controlador*, haciendo de intermediario entre ambos, gestiona el flujo de información realizando las transformaciones necesarias.

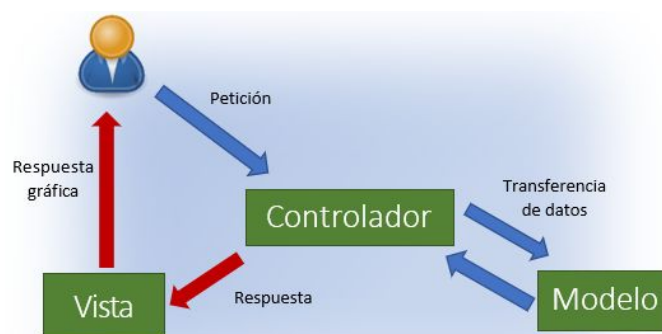


Figura 6.2: Representación visual del modelo-vista-controlador



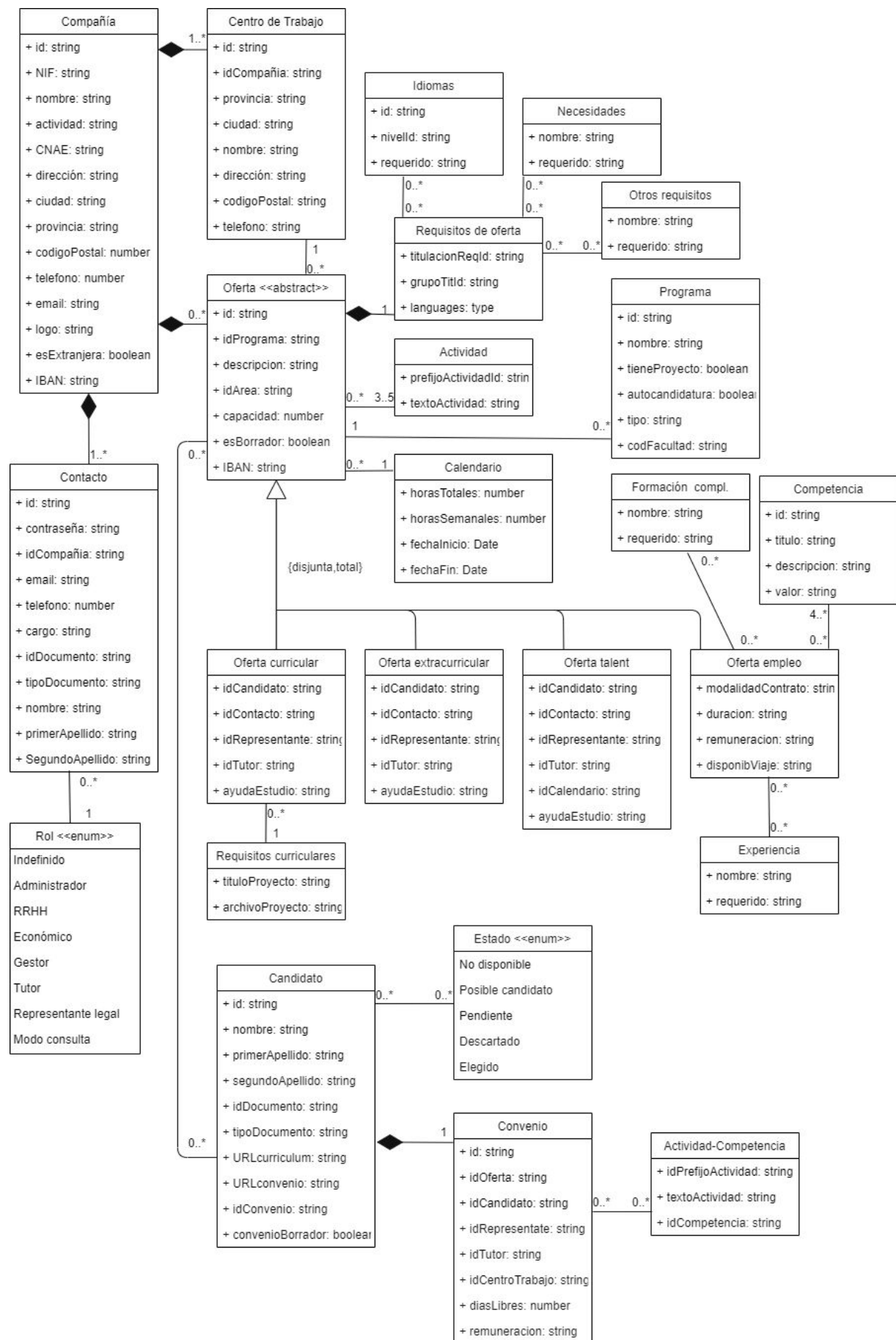


Figura 6.3: Diagrama de clases con datos manejados en la aplicación

La *vista*, o capa frontal de la aplicación se ha implementado con el framework Angular 8. El *modelo* está recogido en una ERP Navision. El *controlador* se ha implementado mediante un servicio web que encapsula la lógica de negocio, solicita los datos necesarios a los servicios de almacenamiento de la ERP Navision y los ofrece a la vista. El servicio web valida las peticiones del usuario procedentes de la vista trasladándolos a Navision. También transforma los datos recibidos desde Navision al formato esperado por la vista. Entre la lógica de negocio incorporada en los servicios web está la gestión de permisos del usuario, es decir la comprobación de que el usuario puede acceder a los contenidos o acciones que solicita. Para el desarrollo de la lógica de negocio se ha utilizado el framework .NET Core 3.1.

## 6.6. Diseño de datos

Los datos que maneja la aplicación proceden de la ERP del cliente. Estos datos llegaban al Frontend por medio de peticiones HTTP al Backend. El Backend, a su vez, obtenía estos datos por medio de peticiones HTTP al servidor del cliente que contiene la base de datos.

La Figura 6.3 muestra el diagrama de clases con la estructura de los datos manejados por la aplicación. Vamos a comentar brevemente las clases principales. La clase *compañía*, una de las de mayor número de atributos, contiene información de las empresas que ofrecen prácticas. Hay bastantes clases alrededor de esta, la clase *contacto* contiene a las personas que tutorizan a los estudiantes, la clase *oferta* las prácticas ofertadas. Aunque quizá no quede muy claro inicialmente, la clase *contacto* contiene a todos los usuarios de la aplicación, incluyendo a los administradores de la universidad. Estos últimos están asociados a la empresa especial “Fundación universidad-sociedad”. Se diferencian entre unos y otros usuarios por el valor de *rol* (administrador, tutor, modo consulta...). Se ven también los cuatro tipos de oferta representados en las clases *oferta curricular*, *extracurricular*, *talento* (talent) y *empleo*. Otra entidad importante es *candidato*, correspondiente a un estudiante de la universidad.

## 7. Implementación y verificación

Aquí se explicará la parte más tecnológica relacionada con la programación y la verificación y puesta a punto posterior (realizada con la participación del cliente), y con las herramientas de dirección del proyecto.

En siguiente subsección se presentarán las herramientas y tecnologías utilizadas para estas actividades. Recordemos que el TFG asumió parte de las tareas de un proyecto ya comenzado. Por tanto, la selección de herramientas se había hecho con anterioridad y hubo que adaptarse a ellas. Después se presentan, en sucesivas subsecciones, partes del proyecto donde se ha intervenido en el TFG y destacando algunas dificultades encontradas y cómo se resolvieron.



## 7.1. Tecnologías y herramientas

En el TFG se han utilizado varias herramientas de dirección de proyectos y de desarrollo de aplicaciones. Las vamos a dividir según su orientación:

1. **Entornos de desarrollo:** Se utilizó **Visual Studio** para desarrollar la lógica de la aplicación (*backend*) y **Visual Studio Code** para la interfaz (*frontend*). También se utilizó **Postman**, un entorno, que se utiliza mediante una extensión de Google Chrome, que permite diseñar, probar y monitorizar servicios HTTP REST sin necesidad de desarrollar un cliente y con el que se probaron los *endpoints* del *backend*.
2. **Control de versiones:** Se ha utilizado **Git** para controlar las versiones de los archivos compartidos en el equipo del proyecto. También se usó el cliente para Git **Fork**, que facilita la tarea ofreciendo una forma más visual de controlar las diferentes ramas.
3. **Seguimiento del proyecto:** Se utilizaron dos herramientas, por una parte **Redmine** donde se recogen los tiempos de dedicación al proyecto. Por otra parte se usó **Trello**, que comparte un tablero kanban que representa las tareas en las que se ha dividido el proyecto y el punto de proceso en que se encuentra cada una (por ejemplo: sin comenzar, programación, prueba, verificación...).
4. **Accesibilidad:** Para comprobar la accesibilidad de las páginas web se usó la herramienta **WAVE** (*Web Accessibility Evaluation Tool*), otra extensión de Chrome.

Respecto a las tecnologías relacionadas con la programación de la solución distinguimos las siguientes:

1. **Frameworks:** se utilizó **Angular 8** para desarrollar el *frontend* de la aplicación y **.NET Core 3.1** para el *backend*.
2. **APIs:** se utilizó la librería **Bootstrap** como complemento al diseño HTML y CSS de la aplicación web. También se utilizó la plataforma de servicios web **IIS** (*Microsoft Internet Information Services*) para publicar en internet las páginas web.

## 7.2. Reglas de validación de datos

Nos referimos en esta sección al establecimiento de reglas de validación en los campos de los formularios, especificando los criterios a cumplir por todos los valores introducidos en un determinado campo. Algunos ejemplos son forzar a “introducir algún valor”, “un número positivo”, “que se cumpla una expresión regular”, etc. Esta validación se hace en la interfaz cuando el usuario introduce o modifica datos de los campos.

Para esta actividad se utilizaron las reglas de validación de Angular. Estas reglas se asocian a campos individuales pero no se incluye ninguna que permita comparar varios campos del formulario o lanzar consultas, por ejemplo a la BD, para comparar después su resultado con el valor introducido en el campo. Destacamos dos páginas en las que encontramos **dificultades**:

1. En **cambiar contraseña** necesitábamos comprobar que los campos “nueva contraseña” y “verificar nueva contraseña” tuviesen el mismo valor.
2. En **crear nueva oferta** hacía falta comprobar tres cosas. Primero que en los campos “titulación requerida” y “grupo de titulaciones” sólo uno tenía un valor (y el otro quedaba vacío). Segundo, que el campo “horas semanales” contenía un valor inferior

a 40. Tercero, que el valor de “fecha de inicio estimada” era anterior al de “fecha de fin estimada”.

La **solución** a todos estos problemas fue programar funciones especiales para realizar cada validación no contemplada por Angular, añadiéndolas después como validadores del formulario. Así, se programó una función que devuelve un error si los campos “nueva contraseña” y “verificar nueva contraseña” contienen valores diferentes, en los campos “titulación requerida” y “grupo de titulaciones” se creó otra función que devuelve un error si ambos contienen valores no nulos. En “horas semanales” se creó una tercera función que devuelve error si contiene un valor mayor a 40, y una cuarta que da error si “fecha de inicio estimada” contiene un valor mayor que el de “fecha de fin estimada”. La misma idea se aplicó a otras páginas donde, por ejemplo se necesitaba comprobar el formato de NIF, DNI o pasaporte en un mismo valor dependiendo de la selección hecha en otro campo.

### 7.3. Errores del *backend*

Además de los errores inherentes a los datos introducidos mediante la interfaz, se pueden producir otros errores debidos a problemas con el hardware, o los servidores. Por ejemplo, fallos de memoria o denegaciones de servicio (el servidor de BD no responde). También pueden lanzarse errores desde la lógica de negocio. Por ejemplo si los datos aportados para el inicio de sesión (usuario y contraseña) no corresponden a ningún usuarios registrado. Otro ejemplo similar es aportar datos de una nueva empresa pero indicando un NIF que ya existe en la BD. En cualquiera de los casos el *backend* va a enviar un mensaje de error a la interfaz y es preciso que la aplicación continúe en funcionamiento.

La **solución** a estas circunstancias es la utilización de objetos Angular de tipo TemplateRef. Se trata de ventanas que muestran un mensaje al usuario. Como todos los errores lanzados por el Backend provienen de peticiones HTTP, el texto mostrado en dicha ventana es el propio mensaje de error recibido.

### 7.4. Multilingüismo

La solución web debía mostrarse indistintamente en español, euskara e inglés. La opción de cambio de idioma se situó en el menú lateral y se construyó una solución alrededor del servicio TranslatorService de Angular. Se construye para cada idioma un fichero JSON con una lista clave-valor que asigna el valor o texto correspondiente a cada elemento de la interfaz, como se ve en el siguiente ejemplo:

fichero ES.json:	fichero EN.json:
<pre>{   "login.signup": "Iniciar sesión",   "login.logout": "Cerrar sesión",   "login.passRecover":     "¿Has olvidado tu contraseña?" }</pre>	<pre>{   "login.signup": "Log in",   "login.logout": "Log out",   "login.passRecover":     "Did you forget your password?" }</pre>

En HTML leemos lo siguiente: `<label> {{ 'login.signup' | translate }} </label>`, que será sustituido por el texto “Iniciar sesión” o “Login” del JSON correspondiente. Pero la **difícultad** es cómo indicar cuál de los tres ficheros JSON debe considerar el HTML en cada momento.

La **solución** consistió en modificar el servicio “`translator.service`” de Angular, que asigna por defecto a `ES.json` como fichero a considerar. Se añadió a dicho servicio una función que recibe como parámetro el idioma actual y activa el JSON correspondiente. Los botones del menú lateral invocan a dicha función de “`translator.service`”.

## 7.5. Verificación y depuración

Como ya se ha comentado, durante la realización del TFG el cliente universitario realizó la verificación exhaustiva del producto. Esta es una de las últimas etapas del proyecto en el que se enmarca el TFG. Esta es una de las tareas más importantes del TFG, y exige la depuración de una lista de errores detectados en la aplicación y una lista de cambios solicitados sobre un producto construido mayoritariamente por el equipo mallorquín. La ejecución de estas tareas se ajustó al calendario previsto (diagrama de hitos de la Figura 4.1).

Para ilustrar esta actividad, se explican a continuación las intervenciones más comunes:

- **Inclusión de campos a formularios.** Estos cambios afectaron también al *backend*, ya que afectan a la estructura de los objetos de la aplicación. Por ejemplo, se pidió separar el campo apellidos de un contacto en primer y segundo apellido. También se cambió el tipo de documento que se manejaba, inicialmente era un DNI, y después se pidió que se distinguiera entre DNI, NIF y pasaporte separando la información en tipo de documento y número.
- **Restricción de funcionalidades a roles.** Por ejemplo, se pidió que solo pudiesen crear contactos, ofertas, etc... los usuarios con el rol de administrador. Al principio se había planteado que los contactos tuvieran roles diferentes, pero no se llegó a establecer qué funcionalidades podía ejecutar cada rol. Se tuvo que ajustar la aplicación para que algunas páginas fuesen accesibles exclusivamente por administradores.
- **Inclusión de ayudas emergentes (*tooltips*).** Los clientes solicitaron la incorporación de este tipo de aclaraciones en algunos campos de la interfaz. Por ejemplo, en el campo ‘Otros requisitos’ de las ofertas se incluyó un tooltip con el texto “Carnet de conducir, vehículos, etc...”. Lógicamente, estos tooltips cambian según el idioma seleccionado en la interfaz.
- **Verificación de la accesibilidad de nivel AA.** Esto incluye a todas las páginas de la aplicación. En algún caso hubo que corregir algún detalle como campos de entrada sin etiqueta asignada o botones sin texto.
- **Interfaces móviles.** También hubo que revisar el funcionamiento de la aplicación desde tabletas y smartphones. La mayoría de las páginas de la aplicación tienen elementos responsivos, es decir que se adaptan al tamaño del dispositivo, gracias al uso de la librería Bootstrap (HTML y CSS). Sin embargo, en algunos casos Bootstrap no se ajustaba bien al tamaño disponible. Esto pasó, por ejemplo, en un smartphone al mostrar una tabla que contenía horas (del formulario de ofertas de tipo talento). Así que se modificó una variable para cambiar su aspecto.

## 7.6. Una página especial: “modo consulta”

Como ya se ha mencionado, la última página que se desarrolló en el TFG fue la denominada “modo consulta”. Solamente está disponible para usuarios con el rol especial habilitado para acceder a esta página. En ella se muestra un desplegable con todas las compañías y una vez seleccionada una, se pueden consultar sus contactos y ofertas. Sin embargo, no se puede modificar ninguno de sus datos.

Inicialmente la compañía que se podía consultar era exclusivamente la del usuario de la sesión, así que, la **dificultad** estaba en la forma de acceder con un usuario, perteneciente a una compañía, a datos de otra compañía diferente.

La **solución** consistió en almacenar en los datos de la sesión cuál es compañía de la que puede obtener información. Por defecto esta compañía es la que corresponde al usuario que inicia la sesión, pero en los usuarios con el rol de modo consulta, cuando seleccionan otra compañía en la pantalla, se lleva a los datos de sesión sustituyendo la anterior.

## 8. Implantación

Al tratarse de una aplicación web, a ser utilizada por el cliente, es necesario desplegarla en uno de sus servidores que disponen de Microsoft IIS y configurarla para habilitar el acceso público. Durante la ejecución del proyecto se han desplegado varias versiones de la aplicación. El Proceso de despliegue consiste en dar los siguientes pasos:

1. Publicación del *frontend*, ejecutando la instrucción “ng build” desde Visual Studio Code.
2. Publicación del *backend*, ubicado en una carpeta, con la opción “publicar” de Visual Studio.
3. Compartir los archivos anteriores, conectándose a un equipo ubicado dentro de la red del servidor del cliente, mediante la aplicación TeamViewer.
4. Copiar en el servidor el proyecto publicado dentro de las carpetas asignadas a los sitios web para el Backend y el Frontend de la aplicación. Esto se hace conectándose al servidor del cliente mediante el escritorio remoto de Windows. Allí se copia el proyecto publicado dentro de las carpetas asignadas a los sitios web para el Backend y el Frontend de la aplicación.

## 9. Conclusiones

Este TFG tiene dos peculiaridades sobre las que merece la pena hacer una reflexión en forma de lecciones aprendidas. La primera tiene que ver con la tarea de *verificación general de la aplicación*. Se hace en estrecha colaboración con el cliente y consume una cantidad importante de tiempo. Exige la comprensión en profundidad tanto de los requisitos funcionales como del diseño e implementación realizadas por el equipo. El éxito de esta tarea debe mucho a conseguir la colaboración constante del cliente. La interacción no se rompió después de la reuniones iniciales de recopilación de requisitos, sino que se realizó un

prototipo donde contrastar lo que se había entendido. En segundo lugar, el desarrollo iterativo ha ido generando versiones, que se desplegaban y se interactuaba con el cliente mediante demostraciones. Por último se ha hecho una última revisión general del producto. En todas las interacciones con el cliente se han ido resolviendo problemas, y añadiendo mejoras. Esta constante comunicación ha evitado la aparición de necesidades ocultas que podrían haber atrasado la entrega del producto.

La segunda peculiaridad que merece una reflexión en forma de lección aprendida es la *integración con equipos virtuales*, es decir situados en lugares geográficos lejanos, que exigen la colaboración mediante medios telemáticos. Cuando los miembros del equipo de desarrollo trabajan en el mismo espacio y sus puestos están muy cerca unos de otros, la comunicación mejora notablemente. En nuestro caso las herramientas telemáticas nos han hecho consumir más tiempo. Cuesta sincronizar, conectar, a veces la línea se interrumpe, etc. Sin embargo, el método seguido nos ha dado muy buenos resultados y creo que es al menos un principio de modelo de telecolaboración. Las reuniones cortas diarias por videoconferencia, al estilo de los *briefings*, la recopilación de avances y decisiones por email, la utilización constante del chat para dudas de menor calado y el acceso a la última versión de toda la documentación del proyecto (y a las versiones anteriores gestionadas por la propia herramienta), nos ha permitido estar informados constantemente del estado en el que se encontraba el proyecto. Al menos para un equipo de desarrollo de cinco a siete personas parece una opción muy interesante y muy diferente a las que utilizamos en equipos de tamaño similar de la asignatura de proyectos donde la comunicación se centraba en la mensajería de texto de whatsapp.

La incorporación a un equipo, que estaba desarrollando un proyecto real para un cliente, ha servido para profundizar y complementar los contenidos de las asignaturas del grado. A continuación incluyo un par de reflexiones sobre lo aprendido durante mi experiencia realizando el TFG en la empresa.

Un aspecto fundamental para entender bien las necesidades y solicitudes del cliente ha sido la *realización de un prototipo*. Obviamente exigió una inversión considerable de tiempo, pero los beneficios conseguidos en comprensión y corrección de los requisitos permitió tener una visión muy clara de lo que pedía el cliente. Como ya hemos dicho, el prototipo simplificó la verificación general del producto, una de las principales tareas del TFG.

Este proyecto tenía una fecha de entrega importante para el cliente. Por esa razón he podido observar cómo se aplicaba la *triple restricción*, un concepto conocido de dirección de proyectos que relaciona el tiempo de realización del proyecto con su alcance, coste y calidad. Una vez se fijó el alcance y se matizó, tras la revisión del prototipo junto al cliente, se constató que no se podría terminar para la fecha fijada únicamente con el personal disponible para el proyecto en Logroño. Por tanto, se incrementó el número de trabajadores (coste) incorporando primero personal de Mallorca y más adelante de Zaragoza. También he observado que se manejan márgenes de tiempo más amplios de los planificados por si surgen dificultades inesperadas durante el proyecto. Esto es así a pesar de que la empresa tiene mucha experiencia con la tecnología que maneja.

La realización del TFG también ha servido para profundizar en el conocimiento de tecnologías destacables de desarrollo web como Angular y API REST por medio de C#.